Graduating to Postdoc: Information-sharing in support of organizational structures and needs

Richard M. Keller[1] Paul J. Lucas[2] Michael M. Compton[3]
Helen J. Stewart[1] Vinod Baya[2] Martha Del Alto[2]
NASA Ames Research Center
Computational Sciences Division
Moffett Field, CA 94035-1000 USA
+1 650 604 3388
keller@ptolemy.arc.nasa.gov

---

[1]National Aeronautics and Space Administration
[2] Caelum Research Corporation
[3] Recom Technologies, Inc.

ABSTRACT
The deployment of information-sharing systems in large organizations can significantly impact existing policies and procedures with regard to authority and control over information. Unless information-sharing systems explicitly support organizational structures and needs, these systems will be rejected summarily. The Postdoc system is a deployed Web-based information-sharing system created specifically to address organizational needs. Postdoc contains various organizational support features including a shared, globally navigable document space, as well as specialized access control, distributed administration, and mailing list features built around the key notion of hierarchical group structures. We review successes and difficulties in supporting organizational needs with Postdoc.

Keywords
Document-sharing, access control, user groups, organizational groupware

INTRODUCTION
With the advent of the World Wide Web, the use of document-sharing and information exchange systems to facilitate communication and collaboration is burgeoning within large organizations. Pioneered by Lotus Notes [12] and BSCW [1-4], many new systems have been developed in the commercial sector and are quickly infiltrating the marketplace. Some of these new systems include DocuShare [20], Livelink Internet [16], Virtual File Cabinet [13], and RightSite [10]. Web-based delivery makes these systems more viable and more affordable for large organizations, given the ubiquitous presence of Web clients.

For the most part, the technical challenges of providing cross-platform, asynchronous document-sharing and information exchange via the Web are well understood. Less certain is how to design and engineer these systems appropriately for widespread and effective organizational use. For information-sharing systems to gain acceptance, they must support organizational needs and structures, and must respect existing organizational patterns of communication. Without careful design, information-sharing systems have the potential to disturb the organizational balance along two central axes: information control (i.e., what information is available, and to whom) and organizational authority (i.e., how is information control managed).

Information is a valuable organizational resource, and the responsibility either to disseminate or withhold information is delegated to individuals with appropriate oversight and authority within the organization. Traditionally, organizational authority is delegated in accordance with the prevailing management structure as set forth in the organization chart, and information control is guided by a set of policies and procedures. Different organizations can be characterized as being at different places in the space of information control and organizational authority. At one extreme, an organization may exert tight control over information access and grant exclusive authority for information control to a small set of individuals in line management. By contrast, another organization may support widely open information access and grant everyone equal authority to control access. Most organizations lie between these extremes.

While there is a certain need to exert information control and organizational authority, a successful organization balances these needs against the benefits of supporting open access to information and looser lines of authority. For example, many large corporations have acknowledged the value of sharing information resources, and have invested heavily in the development of knowledge management infrastructure [9, 15]. The benefits of information-sharing include reduced duplication of effort, increased productivity, and enhanced organizational awareness. Many organizations have also recognized the need to support the formation of *ad hoc* work groups lying outside the normal organizational structure. These groups are established dynamically in response to temporary organizational needs or non-organizational (e.g., social) needs. Some examples include crisis management teams, review panels, hiring committees, coffee clubs, etc. The ability to establish extra-organizational workgroups and non-conventional lines of authority rapidly as needed

enhances the organization's responsiveness to unforeseen circumstances.

Computer-based systems can help organizations reap the benefits associated with information sharing. However, there is a significant potential downside to the technological solution from the organizational perspective. In particular, information-sharing systems have the potential to bypass established organizational mechanisms for controlling and authorizing information access. If properly designed, however, information-sharing systems can support rather than subvert organizational goals. By building systems with sufficiently flexible organizational support mechanisms, it is possible to configure a system to meet the specific information access and organizational authority balance appropriate to the organization. An organization's legitimate need to adjust that balance must be acknowledged and supported by the software.

Some organizational requirements for information-sharing include:

- *Universal access*: Members of the organization must be able to access information from multiple computing platforms, whenever and wherever necessary to support their work, whether in the office, at home, or on travel.
- *Flexible security and access control mechanisms*: In many organizations, security and control of sensitive information is the primary concern when adopting an information-sharing system.
- *Support for organizational structures*: Typically, information-sharing policies within organizations are based on the management structure. Information-sharing systems must function in the context of various types of management structures, including hierarchical and matrixed structures.
- *Support for organizational change*: Change is an ongoing process within most large organizations, and information-sharing systems must support various types of changes, including organizational restructuring, personnel changes, modifications in authorization policies, etc.
- *Distribution of authority*: Authority over information-sharing decisions should be flexibly distributed over responsible individuals in the organization, as necessary.
- *Delegation of authority*: Individuals with authority should be able to delegate that authority temporarily to others [8, 18].
- *Support for multiple roles*: Individuals within an organization play multiple roles and have different information needs when assuming different roles.

Without sufficient support for these types of requirements, information-sharing systems will be rejected by organizations as disruptive and counter-productive to organizational goals.

This paper describes the Postdoc system [6], a flexible Web-based document-sharing and information exchange system designed to support many of the organizational information-sharing needs described above. (Note: This paper is based on the version of Postdoc released as of March 1998. Significant enhancements have been introduced since then, and these remain outside the scope of this paper.) Postdoc is fully-implemented and deployed, with an active user base in excess of 500 users. Postdoc contains various specialized organizational support features including a shared, globally navigable document space, as well as access control, distributed administration, and mailing lists features built around the central notion of hierarchical groups structures. The next section describes Postdoc, placing special emphasis on its organizational support features. Following this section, we describe our experience deploying Postdoc within a large organization, and discuss a number of organizational information-sharing issues that came to light as a result of this process.

THE POSTDOC SYSTEM

Overview

Postdoc is a multi-user, web-based application designed for storage, retrieval, and sharing of information (e.g., documents, images, graphics, software, e-mail) within organizational workgroups. Unlike a traditional web site where the webmaster is in charge of the content, a Postdoc web site is constructed by its users who can add, delete, and organize information however they want. Users need not have any specialized web site construction knowledge or skills such as HTML or programming. Using application software, users create files on their own computer in a variety of formats (including Microsoft® Office documents, Adobe® Photoshop images, MPEG movies, and many others) and then upload them to the Postdoc server after initiating a password-protected login. Depending on the file format, the server can then automatically convert files into Adobe's PDF (Portable Document Format) enabling them to be viewed without the original application using Adobe's free Acrobat Reader. By virtue of being a Web-based application, Postdoc can be used by anyone, anywhere, having a computer (be it a Macintosh®, PC, or a Unix workstation), a web browser (such as Netscape® Navigator or Microsoft® Internet Explorer), and an Internet connection. Postdoc works well for small, co-located teams, as well as geographically dispersed teams including members on travel using a laptop computer with a cellular modem. Other major Postdoc features include the following :

- full-text indexing and searching for documents in selected formats
- security and access control mechanisms*
- hierarchical user groups*
- distributed administration*
- mail services, including mailing lists* and threaded e-mail archives
- subscription and notification
- document revision control

• extensive on-line help and documentation

In this section, we will focus on features of Postdoc that specifically address organizational needs, including those items with an asterisk in the list above (for others features see [6]).

The Postdoc system described in this paper is actually a second-generation system built to improve upon the capabilities of a document-sharing system originally designed to support collaborative work between NASA Ames Research Center in the San Francisco Bay area and the Jet Propulsion Lab in the greater Los Angeles area. The original system -- called the New Millennium Program Electronic Documentation System (NMP-Doc) [5] -- was designed and built shortly after the introduction of the Web, and was released to users in June 1995. NMP-Doc still supports over 700 users exchanging documents, data, schedules, presentations, and software related to NASA's New Millennium Spacecraft Project. Postdoc, which was released in March 1997, was designed to be more portable and to provide access controls not available in the original system. The hierarchical user group features highlighted in this paper were added in September 1997.

Global Document Space

Each Postdoc server manages a global document space that is presented to the user as a conventional "desktop" folder and file display. The document space is organized as a tree structure with a single, all-enclosing folder at the top level. All documents stored on a given server can be accessed by traversing hierarchically downward from the set of document folders within the top-level folder. The structure of the document tree underneath a given folder is completely arbitrary and is under the control of the users who "own" the folder.

Figure 1 illustrates the top-level document space for a Postdoc server hosted by the fictional Acme Industries. The top-level "Documents" folder contains a set of folders containing information pertinent to the various companies owned by Acme Industries. In addition, the top-level folder contains individual items relevant to the parent company and its holdings. In general, folders can contain four different types of items:

- *Documents*: application files in various formats, including word processing documents, spreadsheets, presentations, images, graphics, text files, and file archives. Certain document types (e.g., Microsoft Office documents) are automatically translated to cross-platform PDF format, and displayed along with the native format. Depending on the user's Web browser settings, clicking on a document either launches the helper application associated with the document's MIME type or saves the document to the user's local disk;
- *Notes*: short text notes, which are displayed inline within the folder;
- *Links*: URLs for items on the Postdoc server or arbitrary external pages. Clicking on a link displays the linked Postdoc page or external Web page;
- *Folders*: embedded containers for sets of items. Clicking on a folder causes the Postdoc interface to descend into the folder and will result in a display of its contents.

Figure 1: Top-level "Documents" folder

Users can navigate through document space by clicking on folders to descend in the hierarchy and clicking on the "Up" link displayed within each embedded folder to climb the hierarchy. In addition, users can activate a separate navigation window displaying the hierarchy by clicking on the "Folder Hierarchy" icon displayed at the top of each folder display. This window presents a compressed, nested folder display for the folders above and below the current folder. Clicking on a folder in this window displays the folder contents in the main Postdoc browser window.

Hierarchical User Groups

For purposes of accessing the document space, users are organized into sets of hierarchically-related groups. Overall, the groups form a directed acyclic graph. Each group can have a set of parent groups and a set of child groups. (Note that this group structure is separate and distinct from the hierarchical document structure.) Figure 2 depicts a portion of a typical group structure. Each node represents a group consisting of a set of individual users. For example, the group labeled *Sales Division* consists of individuals directly involved in Division management functions. The group has one parent (*Acme Widget*

Company) and two siblings (*Information Services Division* and *Products Division*). The child groups under the Sales Division include various departments at the next level and projects underneath that. The bulk of the Sales Division's personnel with be members of groups at the lower project levels. Note that *Web-based Sales Project* is a subgroup of both the Information Services Division and the Sales Division.

Users can view the members of any group, as well as its "parent" and "child" groups by accessing the group display functions (see Figure 3). In addition, users can display and navigate the group hierarchy in a separate window, as with the document hierarchy.

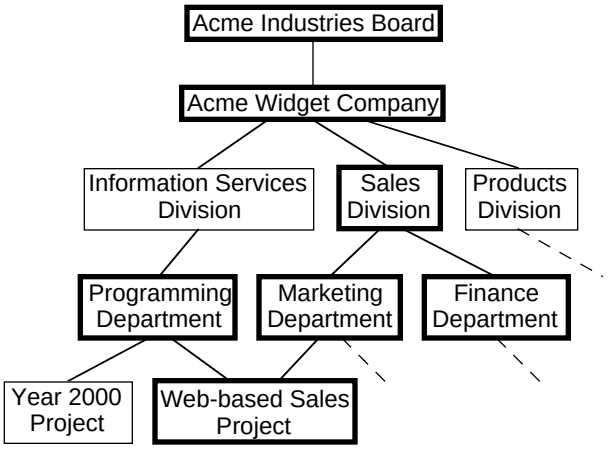

Figure 2: Hierarchical group structure

Figure 3: Group management page

Each group has one or more individuals designated as "owner". The owners of a group have the responsibility for administering the group, which involves managing the group's membership list and fielding requests for new membership. When a new user registers to use the Postdoc server, that user specifies a primary membership group. The owners of this group are automatically notified and have the responsibility for approving or declining the new user's registration and membership request. Once a user's registration and primary membership have been approved, the user is notified via e-mail, and can request membership in other groups on the server. Group owners can add or delete any registered user from their group at any time. Any registered user can create a new group and become its owner. As owner, the user can install any *existing* group as a parent or child of the new group, but this requires the permission of the respective owners. (Cycles in the group hierarchy are disallowed.) In comparison with centralized registration and membership mechanisms, this type of group mechanism distributes the burden of site administration across a larger set of people and places the control in the hands of the appropriate people -- the group owners.

Aside from the notion of group membership, there are other useful group-related constructs defined in Postdoc. If we take the union of all the individuals belonging to subgroups beneath the Sales Division group (i.e., the direct subgroups of Sales Division, their children, their children's children, and so on), these individuals belong to Sales Division's *extended child group*. In this example, the Sales Division's extended child group contains all individuals within the department-level and project-level groups underneath the base Sales Division group. Note that this definition of hierarchical user groups differs from the definition given by both Sikkel [18] and Shen and Dewan [17]. In these alternative approaches, group membership corresponds to the *union* of the base group and extended child group membership in our formulation. Our restricted notion of group gives us an additional distinction that proves useful for granting access rights. Intuitively, it is sometimes necessary to grant rights to the base group (the Division management personnel) that would be inappropriate for the extended child group (the Division workers).

Another group-related construct utilized in Postdoc is the notion of an *extended parent group*. The extended parent group for a group consists of the union of all users belonging to its ancestor groups (its immediate parent group, its parent's parents, and so on up to the group's roots in the group structure). Intuitively, it is sometimes useful to consider all individuals above a group in the hierarchical structure (the Division's superiors) as a consolidated group for purposes of granting access rights.

Access Rights Management

One of the primary uses of the hierarchical group structure within Postdoc is the control of access rights to documents and folders. Figure 4 illustrates how access rights are granted to groups and individual users. The "Item Relationships" page describes the relationships between the item being described (in this case a folder) and either groups or individual users. (In terms of Lampson's

matrix formulation [14], this display illustrates the access control list for the item.) Seven types of relationships are defined in Postdoc:

- *Owner* (O): The group or user owns the item, and thus has the authorization to configure the item's relationships.
- *Author* (A): The group or user is the author of the item. The authors of an item are shown by Postdoc in the folder view (see Figure 1) and certain other display modes.
- *Readable* (R): The group or user can read the item. Group read permits all members of the group to read the item.
- *Writeable* (W): The group or user may modify the item. Group write permits all members of the group to modify the item.
- *Parent Readable* (P): The members of the group's extended parent group can read the item.
- *Child Readable* (C): The members of the group's extended child group can read the item.
- *Subscriber* (S): The group or user is subscribed to the item and receives automatic notification whenever it is modified. Any registered user may subscribe to an item; this relationship can be established by the owner (via the Item Relationships display) or the user (via the Item Relationships "subscribe" button displayed to non-owners).

Figure 4: Item relationships page

In Figure 4, *read* access for the "Divisional Procedures" folder has been granted to the Sales Division group and to both its extended child group and extended parent group. *Read* access has also been temporarily granted to the Programming Department group, which is reviewing the Sales Division procedures as a model for revamping their own procedures. The child groups under Programming cannot access the folder, nor can its parent, the Information Systems Division group. For clarity, all of the groups with access to the Divisional Procedures folder have been marked with a thick border back in Figure 2. Joe User is the owner of the folder and another user has been granted *read*/*write* access, as well. The owner of this folder can grant access rights to other existing groups or registered users by entering a search string in the Find box on the display. All users or groups matching the string are fetched and displayed to the owner, who can establish relationships as desired.

Although the relationships established for an item (a document, note, link, or folder) are strictly independent of the relationships established for its enclosing folder, an item's initial relationship settings upon creation are based on the enclosing folder's settings. If the item added to a folder is *itself* a folder, then the new folder copies the *read*, *write*, *parent read*, and *child read* access privileges from the enclosing folder; if the item is a document, a note, or a link, then the same access privileges are copied, excluding the *write* relationship, which is initialized to non-writeable. This copying is done one time only upon initialization; subsequent changes to the enclosing folder will not affect the access privileges associated with the contained items. (Note: The copying action is explicit at creation time, in contrast with the implicit inheritance of access rights based on hierarchical group structure at the time of item access, as with the *parent read* and *child read* relationships.)

For comparison, consider how the example shown in Figure 4 would be handled by either Sikkel's [18] or Shen and Dewan's [17] group formulations. In these cases, granting read access to the Programming Department group would not only grant access to the members of the Department group, but also to staff members in project-level groups beneath the Department group. To accomplish the same effect as Postdoc achieves with one group, these alternative formulations would require two groups. In particular, one would need to create a second group called *Programming Department Management* -- an unrelated sibling of the Programming Department group. This group would duplicate all the members of the Programming Department group, but would have no child groups. Granting read access to the Programming Department Management group would accomplish the goal of blocking access to the non-management staff. This solution is awkward because it introduces redundancy and creates problems associated with keeping the individual members of the two groups in synch.

Mailing List Administration

Almost as a fringe benefit, a hierarchical group model works nicely with electronic mailing lists. Whenever a group is created in Postdoc, an associated mailing list is also created. (Mailing lists can also be created

independently of groups.) The owners of a group are also the owners of the associated list. Mailing lists in Postdoc have all the features of typical Majordomo-style mailing lists [11] except that their set-up is automatic and their administration is done via a web interface. Additionally, all the e-mail is threaded, archived, indexed, searchable, and accessible via a web interface.

The group hierarchy comes into play in that sending an e-mail message to a group not only sends a copy to every member of that group, but the message also *cascades* down to members of the extended child group as well. A common scenario in which this is useful is for high-level managers to send e-mail to all the people under them. (If managers want to have private correspondence among themselves and not have their e-mail cascade, then they can create a plain mailing list.)

A mailing list in Postdoc can be configured to allow its archive to be accessed only by its current set of subscribers. In light of the cascading mechanism, the set of subscribers is considered to include the extended child group. This makes sense because if users received e-mail initially from a mailing list, then they should be permitted to access its archive.

Dynamic Workspaces

The concept of a group workspace within Postdoc is quite different in character from the notion of a shared workspace in systems like BSCW. In BSCW, a separate and independent document space is created for each workgroup. Users must explicitly shift from one workspace to another to access documents in a different workspace. If the same document is referenced by two workgroups, it must be replicated and carried across to the second workspace. In contrast, Postdoc contains a single global document space and different workgroups have access to different portions of the space. A *workspace* for a Postdoc group can be defined as the set of folders containing documents for which the group has read access. Workspaces for different groups therefore intersect and overlay each other wherever multiple groups have read access to items in the same folder. This notion of workspace is dynamic, rather than static, because it depends upon the permissions at the time of access.

Postdoc's dynamic workspace model has some distinct advantages over the more static and independent model of workspaces in BSCW. With the dynamic workspace model, shifting between spaces is seamless; the user simply navigates the document hierarchy to reach a folder in the desired workspace. This seems to work very well for users who function as part of many workgroups, and allows these users to view their personal workspace as the union of the workspaces in which they are members. More generally, the notion of a global document space increases organizational awareness and enables people in the organization to place their work in the context of other work being conducted by other members of the organization. Another advantage of the dynamic workspace model is that it is easy to temporarily expand a group's workspace as needs arise. For example, during a period of joint collaboration with another workgroup, it may be necessary for group members to access folders not normally within the group's workspace (as with example of the Programming Department group members accessing the Sales Division's "Divisional Procedures" folder in Figure 4). This can be accomplished by temporarily granting the group access rights to the folder in question and rescinding those rights after the work is complete. Finally, the dynamic workspace model allows users to flexibly organize their workspace to cut across traditional hierarchical organization lines. Rather than restrict the workspace to one subtree within the document hierarchy, groups can include components from throughout the global document space in their workgroup as necessary.

Security and Guest Access

As a Web-based application, Postdoc is first subject to the usual security features of Web servers (domain and IP address checks, logins and passwords, etc.). However, the standard login and password mechanism is limited and insufficient for many Postdoc needs. For example, the standard mechanism is based on directories, not files. Also, there is no way to logout or become a different user without quitting the browser. Postdoc therefore then uses its own logins and passwords, and sets a cookie to identify a user. To enable users who are not registered with Postdoc to see certain items on the server, guest user access is provided. By default, users accessing Postdoc prior to logging into the system are considered to be a special user: "Guest User". Guest User has restricted capabilities, but can be added to the Item Relationships page (Figure 4) for any item. The item's owner can grant any set of permissions to Guest User, just as with a regular user.

Summary of Postdoc Organizational Support Features

Postdoc supports many, but not all, of the organizational requirements set forth in the introduction to this paper:

- *Universal access*: Provided by a combination of Web-based delivery, automated document translation services, and Adobe PDF.
- *Flexible security and access control mechanisms*: Provided by a combination of standard Web browser authentication, cookie-based password login, and fine-grained document access control mechanisms.
- *Support for organizational structures*: Provided by user group hierarchy and associated mechanisms for cascading mailing lists and extended parent and child groupings.
- *Support for organizational change*: Provided by group administration and group hierarchy-editing facilities: group owners can approve new membership in groups, and can deleted personnel from groups upon transfer or separation; groups can be unlinked from one place in the hierarchy and relinked elsewhere.
- *Distribution of authority*: Provided by group and item ownership mechanism. Group owners are responsible for managing group membership and associated responsibilities (approval of new users, group hierarchy

changes, subscription requests, etc.). Owners of items (documents, folders, notes, links) are given responsibility for establishing and modifying access permissions.

- *Delegation of authority*: Not provided in Postdoc.
- *Support for multiple roles*: Rudimentary support provided by the user group mechanism. Sets of users performing the same role can be grouped together and given common access rights to items in the Postdoc server. A single user can play multiple roles by belonging to multiple groups. However, there is no mechanism established to enable a user to explicitly assume a selected role; all roles are active simultaneously. This is problematic when assuming a role is intended to restrict user actions [18].

EXPERIENCE WITH POSTDOC

User Community

Postdoc has been deployed within several government agencies to begin addressing some of their information-sharing needs. As of the end of March 1998, a total of four Postdoc servers were being hosted by NASA, NIST (the National Institute of Standards), and NRL (the Naval Research Lab). Each server provides services to a variety of organizations within its host agency. In combination, the four servers support over 650 registered users organized into approximately 150 groups. A combined total of over 3500 documents, folders, links, and notes are currently stored on these servers. (Each Postdoc server is operated independently and there are no facilities for cross-server information sharing.)

Usage Modes

Postdoc is being used to support a wide variety of organizational information-sharing and collaborative activities. Some illustrative examples include:

- *Sharing documents and data in support of research activities*: To facilitate research work at NASA, NRL, and NIST, Postdoc is being used to store and share information among both co-located and geographically-distributed research teams.
- *Satisfying managerial and oversight needs*: At NASA, Postdoc is used to increase the information bandwidth between NASA headquarters and various NASA research and operations centers. NASA headquarters managers require access to information about project status, future plans, and requirements of the programs being conducted at the NASA centers. Postdoc is being used to provide managers with a window into the remote project activity.
- *Managing conference submissions and reviews*: Postdoc has been used to organize conferences and workshops, and to manage paper review processes.
- *Supporting teleconferences*: The system has been used to organize materials for teleconferences and provide participants with simultaneous access to materials during the teleconference.
- *Proposal-writing*: Teams of researchers have used Postdoc to write a large, multi-group research proposal.

Because many of our important customers at NASA are scientists and engineers, we are currently investigating possible customizations to Postdoc that would enable the system to support their special information-sharing needs more directly. These special needs include sharing scientific and engineering data and models and visualizing information.

Issues

The design of an organizational information-sharing system poses many difficult design issues with respect to system architecture, human-computer interface, and social factors. Postdoc represents one attempt to address these issues concretely. In this section, we discuss limitations of the system and review some of the more important issues that have come into focus based on our work supporting Postdoc users within government organizations.

Restricted access rights model

In comparison with other access rights models (e.g., [8, 17, 18]), Postdoc supports a fairly limited set of capabilities. Our intent was to limit the choices to the smallest set of capabilities used in most practical situations. We also sought to reduce the complexity of the system interface and simplify user decisions regarding access control. Designing an intuitive interface, especially within the constraints of HTML, can be very difficult when the access model is complex [19].

As an example of simplifying the access model, we chose not to implement parent and child writeable relationships within Postdoc. Our motivation was that usage of *parent read* and *child read* access would be far more common than the corresponding *write* accesses. Rather than cluttering the interface with additional options, we omitted parent/child write. In addition, we wanted to avoid making it too easy for users to unintentionally grant broad write privileges. Instead, users must accomplish parent/child write by explicitly granting write access to subgroups and/or supergroups.

As another example of access model simplification, Postdoc permits only medium-grain control over items (i.e., documents, folders, notes, links) in the system. Permissions can be established at the item level, but not at the sub-item level (i.e., not at the level of document comments, document and folder titles, creation dates, etc.). Although access control at the sub-item level would be useful in selected instances, we felt the added functionality would not justify the complexity inherent in providing a completely general capability.

Another feature we considered, but omitted, from the access model is explicit denial of access -- so-called "negative rights". The use of access policies involving negative rights has been documented by previous case studies [19] and various schemes have been designed to implement denial of rights [17-19]. However, it is our experience that for many cases, negative rights can be implemented simply by constructing groups

corresponding to exception cases. For example, if read access to document D1 is permitted for group X, but denied to Fred, then Fred can be established as a singleton child group of group X. *Read* (but not *child read*) access for group X can be enabled for D1 to exclude Fred; if Fred is to be granted read access to a different document D2, then *child read* can be enabled on D2 for group X. Certainly other methods of granting negative rights are more expressive, but our concern about interface and access model complexity led us away from these methods.

One access control feature that is missing, but clearly would be desirable in Postdoc is delegation [7, 18]. In several cases, we found that group administration functions, including approval of new group members and management of access control, were not being performed by the designated group owner due to other time pressures and organizational responsibilities. Because in Postdoc there is no mechanism for delegating group ownership functions, the solution was to include the designee as a second owner of the group. While this is a partial solution, has the side-effect of granting total authority to a group member who should not be given unrestricted control. A related problem involves server administration. For the purposes of administration, certain Postdoc users are granted superuser status. This is a case where explicit authorization and assumption of a superuser role would be useful for security purposes. Some Postdoc users have questioned the appropriateness of conferring superuser status upon individuals in our development group, who can view confidential documents as a consequence.

Despite our omission of features and attempt to simplify the access control model, some Postdoc users still find the current scheme confusing. The main source of the confusion lies in the distinction between the document hierarchy and the user group hierarchy. Users think of access rights as being granted based on location in the document hierarchy rather than the group hierarchy. They tend to confuse the concepts of parent group and child group with the parent and child folders in the document tree.

Sensitivity to formal/informal group membership

An interesting side-effect of setting up the group hierarchy within an organization is that it causes people to be explicit about the relationships among workgroups in a highly visible way. Drawing out the organizational structure can be controversial for informal or *ad hoc* groups, where the relationships may be unstated due to rivalry or other political issues. Such sensitivities are highlighted when made explicit within Postdoc. Decisions about group membership and access rights can become the focal point for organizational tensions around authority issues.

Visibility of protected information

One particularly controversial Postdoc design decision involves the way in which non-readable folders and documents are treated by the system. Even though access may be denied, users still can see the titles of all items in folder views and hierarchical navigation views and as part of the search results display for document searches. Locked folders, in particular, are marked with a belted folder icon, though the title is visible (see Figure 1). Our decision to allow the viewing of locked item titles was intended to encourage the sharing of information by allowing users to request access from the appropriate owner. Clearly, some users would prefer a stricter notion of privacy -- one that could be supported by sub-item access control over titles, as discussed above.

CONCLUSIONS

Based on our experience with the broad and diverse Postdoc user community, we feel more strongly than ever that the specialized requirements of organizations must be taken into account when building information-sharing systems. Postdoc provides key support in the form of hierarchical groups and associated mechanisms, including distributed administration, access control via parent and child read permissions, dynamic workspaces, and cascading mailing lists. These mechanisms reduce the burden of controlling users and their information access because the mechanisms are integrally linked with the explicit organizational structure represented by the group hierarchy. Without the benefit of such a structure, organizational information-sharing policies would be tedious to administer and difficult to honor consistently on a system-wide basis. Although many of the difficulties highlighted by more widespread Postdoc usage could be solved with a more sophisticated access control model, our experience suggests that the added complexity of the model and the corresponding user interface could easily overwhelm the user.

ACKNOWLEDGMENTS

This work has been supported by NASA's Cross-cutting Space Technology Program. We wish to thank Keith Swanson for his support of Postdoc and for his initial suggestion to develop hierarchical groups and other organizationally-relevant features. We also want to thank the Postdoc users for their willingness to provide feedback and work with us to gain insight into their needs.

REFERENCES

1. Bentley, R. and W. Appelt. Designing a System for Cooperative Work on the World-Wide Web: Experiences with the BSCW System. in 30th Hawaii International Conference on System Sciences. 1997. Maui, Hawaii:
2. Bentley, R., W. Appelt, U. Busbach, E. Hinrichs, D. Kerr, K. Sikkel, J. Trevor, and G. Woetzel, *Basic Support for Cooperative Work on the World Wide Web*. International Journal of Human Computer Studies, 1997. **46**(6): p. 827-846.
3. Bentley, R., T. Horstmann, K. Sikkel, and J. Trevor. *Supporting Collaborative Information Sharing with the World WIde Web: The BSCW Shared Workspace System*. in *4th International World Wide Web*

Conference. 1995. Boston, MA: O'Reilley & Associates.

4. Bentley, R., T. Horstmann, and J. Trevor, *The World Wide Web as Enabling Technology for CSCW: The Case of BSCW*. Computer Supported Cooperative Work: The Journal of Collaborative Computing, 1997. **6**: p. 111-134.
5. Caelum Research Corp. *New Millennium Program Electronic Documentation System (NMP-Doc)*, http:// nmp-jpl-www. arc. nasa. gov/.
6. Caelum Research Corp. *Postdoc*, http:// ace. arc. nasa. gov/.
7. Coulouris, G. and J. Dollimore. *Requirements for security in cooperative work: two case studies.* Dept. of Computer Science Technical Report # 671, 1994.
8. Coulouris, G. and J. Dollimore. *A Security Model for Cooperative Work.* Dept. of Computer Science Technical Report # 674, 1994.
9. Davis, S. and J. Botkin, *The Coming of Knowledge-Based Business*. Harvard Business Review, 1994. **Sept-Oct**: p. 165-170.
10. Documentum. *RightSite*, http:// www. documentum. com/ rightsite.htm.
11. Great Circle Associates. *Majordomo*, http:// www. greatcircle. com/ majordomo/.
12. IBM. *Lotus Notes*, http:// www. lotus. com/.
13. Infodata. *Virtual File Cabinet (VFC)*, http:// www. infodata. com/.
14. Lampson, B.W., *Protection.* ACM Operating Systems Review, 1974. **8**(1): p. 18-24.
15. Marshall, L., *Facilitating Knowledge Management and Knowledge Sharing: New Opportunities for Information Professionals.* Online, 1997. **21**(5): p. 92-98.
16. Open Text. *Livelink Internet*, http:// www. opentext. com/ livelink/.
17. Shen, H. and P. Dewan. *Access Control for Collaborative Environments*. in *ACM Conference on Computer-Supported Cooperative Work*. 1992. Toronto:
18. Sikkel, K. *A Group-based Authorization Model for Cooperative Systems*. in *European Conference on Computer-Supported Cooperative Work*. 1997. Lancaster: Kluwer.
19. Stiemerling, O. and A.B. Cremers. *A User-centered Approach to Access Control in Collaborative Environments*. in *Second International Workshop on CSCW in Design*. 1997. Bangkok, Thailand:
20. Xerox. *DocuShare*, http:// www. xerox. com/ products/ docushare/ index.html.